Restaurang Unicorn Co

Dokumentation BI-lösning

Innehållsförteckning

[1 Syfte 1](#_Toc5128905)

[1.1 Frågeställningar: 1](#_Toc5128906)

[1.2 Övrigt: 1](#_Toc5128907)

[2 Teknikval 1](#_Toc5128908)

[2.1 Datakällor: 1](#_Toc5128909)

[2.2 Databas 1](#_Toc5128910)

[2.3 Programmeringsspråk 2](#_Toc5128911)

[3 Hur ser lösningen ut? 2](#_Toc5128912)

[4 Förbättringar och utvecklingsmöjligheter 2](#_Toc5128913)

# Syfte

Vi har fått i uppdrag att skapa en BI-lösning för Unicorn Co som vill öppna en ny typ av restaurang någonstans i Sverige baserat på väderdata.

## Frågeställningar:

* *Var någonstans i Sverige* finns det bäst väderförhållanden baserat på flest soltimmar, högst temperatur och minst vind?
* Vilken *typ av restaurangverksamhet* ska Unicorn co fokusera på i sin nya satsning baserat på väderdata, demografidata, konkurrentdata och övrig marknadstrenddata?
* Rekommenderad *prissättning* baserat på: kundgrupp/demografi, konkurrenter?
* Vilken *kundgrupp* ska Unicorn co vända sig till baserat på demografidata?

## Övrigt:

* Föreslå KPIer som verksamheten kan använda sig av när den etablerats.
* Vilka möjligheter finns det med en framtida BI-lösning och en datadriven organisation

# Teknikval

Vi har valt att använda oss av varierade datakällor tex csv-filer som sedan laddats in i ett datawarehouse i SQL där vi tvättat datan och kombinerat den om så behövs. Den tvättade datan har sedan hämtats in i Python för visualisering med hjälp av Bokeh-biblioteket. Vi har även använt oss av webscraping för att hämta in data om konkurrenter. r   
De färdiga visualiseringarna visar vi på en webbsida skriven i html. Se nedan för detaljerad information om teknikvalen.

## Datakällor:

* SMHI
  + Samtliga data är taget för perioden december 2015 – november 2018.
  + Solskensekunder per timme per dag från 19 aktiva väderstationer i Sverige.
  + Genomsnittlig temperatur per månad från de 19 aktiva solskensstationerna.
  + Genomsnittlig vindhastighet m/s per timme per dag från de 19 aktiva solskensstationerna.
  + Samtliga datakällor ovan har aggregerats för att visas i genomsnitt per månad per år. Eftersom att solen är väldigt viktig för Unicorn co kommer restaurangen att vara mest aktiv på sommaren och därmed har vi valt att filtrera ut månaderna maj-september för att få en mer rättvis bild över väderdata baserat på dessa månader.
* Statistiska Centralbyrån
  + Befolkning per region 2018.
  + Restaurangindex per restaurangtyp 2015-2018.
  + Restaurangdata omsättning per restaurangtyp/näringsgren 2017-2018.
* Webscraping
  + Tripadvisor.se

## Databas

Vi har använt SQL Server Management Studio som databas där vi har tankat in vår källdata från SMHI och SCB. Källdatan har aggregerats enligt respektive anvisning under ovan rubrik ”Datakällor” och efter aggregering har denna data exporterats till CSV-filer som vi sedan läst in i Python för vidare programmering.

## Programmeringsspråk

Vi har använt oss av Python som programmeringsspråk med följande bibliotek:

* Bokeh för interaktiva visualiseringar.
* Pandas för att lagra våra dataset i dataframes.

Vi har använt oss av HTML som användargränssnitt. Formatering och design görs i en separat CSS-fil.

# Hur ser lösningen ut?

## Datakällor

* Data som hämtas från Statistiska Centralbyrån laddas ner i en csv-fil eller excel. I den nuvarande versionen av BI-lösningen sparas filen ner i projektmappen ”Data”. Datan hämtas sedan upp i ett Python-script för vidare behandling. *Vi väljer att arbeta med datakällorna lokalt då vi inte hade tillgång till en gemensam SQL-databas inom ramarna för projektet* *men vi utgår ifrån att kunden kommer att ha det. ODBC-drivrutin för Python och script för inläsning finns i ReadME-dokumentationen i GitHub.*
* Data som hämtas in från Tripadvisors hemsida i syfte att kartlägga konkurrenter på orten får vi tillgång till med hjälp av ett webscrapingscript och paketet BeautifulSoup i Python.

## Python-script

* Pythonscripten som behandlar csv-filer och excel-filer innehåller följande delar:
  1. Import av bibliotek tex panda
  2. Kod för inläsning av datakälla
  3. Kod för aggregeringar och omvandlingar om så behövs
  4. Kod för lagring i listor, dictionary, dataframe osv beroende på scrip
  5. Import av Bokehbibliotek
  6. Bokeh API-kod inklusive kod som genererar en HTML-fil med en graf
* Pythonscripten som utför webscraping innehåller följande delar:
  1. Import av bibliotek tex panda samt webscraping-biblioteken BeautifulSoup, Requests, Re och Time
  2. Kod för hämtning av url
  3. Kod för att fylla listor med det innehåll som skrapats från web
  4. Kod för aggregeringar och omvandlingar om så behövs
  5. Kod för omvandling till dataframe beroende på scrip
  6. Import av Bokehbibliotek
  7. Bokeh API-kod inklusive kod som genererar en HTML-fil med en graf

## Visualisering

* Lösningen har en huvudsida som är ”BI-lösningen” som användaren ser i sin webläsare. Den är uppbyggd så att alla individuella grafer som genererats av Python-script länkas in i denna huvudsida.
* All design av sidan styrs från CSS-filen styles.css som länkas till huvudsidan med ett script.

## Struktur

* Alla *datakällor* som lagras lokalt ska sparas ner i mappen ”Data”.
* Alla *Python-script* ska följa namnstandarden MA\_TypAvVisualisering.py alternativt KPI\_TypAvVisualisering och läggas i mappen ”PythonScript”. I det fall att Python-Scriptet genererar en HTML-visualisering ska denna heta samma namn som PythonScriptet men sparas i html-format.
* Alla HTML-filer ska sparas i mappen ”HTML script”.
* All design av HTML-sidan ska göras i CSS-filen styles.css och sparas i mappen ”CSS”.
* Tomma mallar finns att tillgå i mappen ”Blank\_templates”.

# Förbättringar och utvecklingsmöjligheter

I vår lösning har vi lagt till en KPI-dashboard med visualiseringar som verksamheten kan använda för att visa sina KPIer när restaurangen öppnat och riktig data från verksamheten finns att tillgå. Vi ser många utvecklingsmöjligheter vid implementering av en BI-lösning för Unicorn co.  
Med en BI-lösning kan Unicorn co bl.a. få nöjdare gäster, minskade kostnader, optimerade intäkter och bättre planering av personal och tjänster.

Med hjälp av data kan vi:

* identifiera vilka rätter som är populära och inte samt vilka rätter som har bra/mindre bra lönsamhet.
* få större kontroll över kostnader och baserat på detta även ökade nettointäker. Med hjälp av en BI-lösning kan vi se vilka kostnader som är kopplat till en maträtt eller en viss typ av kund.
* få bättre inköpsplanering = mindre svinn.
* ha dynamisk bemanning baserat på data – endast ha så mycket/lite personal som krävs beroende på efterfrågan.
* mäta relevanta KPIer: RevPAS/RevPASH, snittnota, beläggning, bruttovinster, köksprocent.
* jämföra Unicorn co mot konkurrenter med hjälp av t.ex. Benchmarking Alliance Restaurant Trends.
* ta beslut om dynamisk prissättning baserat på data. T.ex. kan meny och priser skilja sig åt beroende på tid, veckodag, råvarukostnader etc.
* erbjuda dashboards i real-tid till personal i service, t.ex. för att hjälpa kockar att planera sina beställningar.
* få skräddarsydda vin-/matrekommendationer med hjälp av AI.
* mäta effekten av erbjudanden
* förlänga livstiden och schemalägga service med hjälp av tex. IOT-sensorer för köksutrustning.
* mäta kundnöjdhet med större precision genom t.ex. Web-scraping av recensioner från externa sidor. Identifiera kundnöjd-/missnöjdhet per dag, plats, rätt och beroende på personal.

Vad gäller den tekniska utvecklingen av lösningen ser vi att den grundstruktur som finns på plats går att återanvända även till de föreslagna nya dataseten som behövs för ovan KPIer. Utveckling skulle dock kunna ske genom att:

* Utöka datamängden som hämtas från Tripadvisor (vi fokuserar på top 30 restauranger) genom att utveckla scriptet så att fler url-sidor läses in.
* Utveckla ett datawarehouse i SQL där all data läses in. Detta gäller då även ny data som genereras i verksamheten så som kassatransaktioner, inköpsdata etc.
* Även bokeh-visualiseringarna kan utvecklas i framtiden. Bokeh har stöd för att bygga fullt interaktiva dashboards genom att använda sk. Widgets. Det är interaktiva kontroller som driver nya uträkningar, uppdatera plottar och koppla samman lösningen med andra program. Detta kan göras med browserns Javascript runtime eller genom att lägga lösningen på Bokehs server.